

CMU 15-445 - Database Systems Lecture 5 - Locks vs. Latches

Locks vs. Latches

Locks and latches mean different things in a DBMS than in an OS.

Locks are - 1. Used to protect the logical contents of the database from other transactions. 2. Held for transaction duration 3. Help to roll back changes

Latches are - 1. Used to protect internal data structures from other threads 2. Held only for operation duration 3. Do not need to be able to roll back changes

Buffer Pool

A buffer pool is just an area in memory that the DBMS reserves to read pages from disk. The buffer pool consists of discrete “slots” of memory we call frames that are the same size as a page. The DBMS reads pages from disk and brings them into frames.

A buffer pool maintains the following metadata - 1. Page table - A hash table mapping page ids to memory location in the buffer pool 2. Dirty flag - A flag indicating whether a page has been modified. 3. Pin counter - The number of threads currently accessing that page. This counter prevents pages that are being accessed from being evicted.

Now that we have introduced buffer pools, we can make the following optimizations - - Multiple buffer pools - We can have multiple buffer pools for different purposes. Each buffer pool can have its own replacement policy and some other properties that may help. For example, having a dedicated buffer pool for each table can help with some particular queries. - Pre-fetching - The DBMS can pre-fetch pages into the buffer pool if it decides that it needs to based on the query plan. The OS can also do this for simple queries, but not in all situations (e.g, with indexes on a query that operates on a certain range of pages). - Scan sharing - If a query is performing a scan over a table, and another query comes in from another thread that also needs

to perform the same scan, it can join the scan that the first query is performing from the page that the scan has reached now.

Replacement Policies

Once we fill our buffer pool, which can happen pretty quickly for large queries, we will need to evict an existing page from the pool to make space for a new page. The way we decide which page to evict is called the replacement policy. This is an extremely important component of the DBMS, because it can make a huge difference in performance. Commercial database systems have a very complex replacement policy as compared to open source systems.

There are a few common replacement policies, but the most popularly used ones are LRU and clock. You already know what LRU is so I won't write about it.

Clock is an approximation of LRU that doesn't need you to store the last accessed time stamps. You maintain a circular queue with a pointer pointing to the "current" page. Each page has a single bit of metadata that indicates whether that page has been accessed since the last time the pointer was here. The pointer always moves forward (or "clockwise"), and when it wants to evict a page, it checks the reference bit for that page. If that bit is 1, that means that the page was accessed fairly recently, and we don't evict that page, but we set its bit to 0. We evict the first page we find that has its reference bit set to 0.